SCIS & ISIS 2010, Dec. 8-12, 2010, Okayama Convention Center, Okayama, Japan

Distribution of Non-dominated Solutions and Preferred Solutions in the Objective Function Space for an Evolutionary Multi-objective Mobile Robot

Yoshiaki Katada
Electrical and Electronic Engineering
Setsunan University
Osaka, JAPAN
Email: katada@ele.setsunan.ac.jp

***Abstract*—Evolutionary multiobjective optimization (EMO) algorithms have attracted much research interest in recent years. In evolutionary robotics (ER), several papers have been published where EMO algorithms have been applied to design multiobjective behavior of autonomous robots. However, these are for either specific control tasks or controllers. Characteristics of EMO algorithms on design of a more popular controller for simple robot control tasks need to be investigated for fully understanding them in ER. In this work, a multiobjective genetic algorithm was applied to the design of a neural controller for multiobjective behavior of a mobile robot in a looping maze problem, which is a popular test problem for ER. Distribution of non-dominated solutions in the objective function space were obtained from a number of trials in the problem in order to investigate how preferred solutions are distributed in them.**

I. Introduction

Evolutionary multiobjective optimization (EMO) algorithms have attracted much research interest in recent years due to their parallel population search [1][2]. Generally, there are two kinds of approach in EMO algorithms; one is an approach to finding the entire Pareto-optimal set (the Pareto-optimal front) taking the advantage of their parallel population search, the other an approach to finding a preferred set of Pareto-optimal solutions, instead of the entire front, based on the decision makers' preference information.

In Evolutionary Robotics (ER)[3] where robot control systems are designed by using evolutionary techniques, most researches employ single objective function to evaluate individuals. In the literatures, we can find several papers dealing with the EMO algorithms in ER [4][5][6]. In those papers, however, it does not seem likely that when artificial neural networks (ANNs), which is a standard controller in ER, are designed, the performance and dynamics of the EMO algorithms are not discussed enough from the methodological point of view .

This paper applied a typical EMO algorithm, NSGA-II[1], to the design of ANNs for a controller of a mobile robot in a looping maze problem, which is a popular test problem for ER, in order to investigate distribution of non-dominated and preferred solutions in the objective function space in the problem. According to the results obtained from the computer simulation, it was discussed which approach might be appropriate in the EMO algorithms mentioned above.

The paper is organized as follows. Section II define a robot control problem where the evolved neural networks are evaluated. Section III formulate the problem as a multiobjective optimization problem. Section IV describes the neural networks adopted in a robot control problem. Section V defines preferred solutions in the problem and gives the results of our computer simulations. Conclusions are given in the last section.

II. Control Task

The control task used in this paper was looping maze, and is based on a task originally implemented by Nolfi & Floreano [3]. Figure 1 shows an experimental setup for this task. The environment of the robot was a square arena surrounded by walls with a circular object placed at the center, which became simpler than the one in [3] for ease of understanding and check experiment. An array of proximity sensors allow the robot to perceive an object and walls. If objects intersect a proximity sensor, the sensor outputs a value inversely proportional to the distance between the object and the robot. A two-wheeled robot was used in this experiment. Employing a mathematical model of a mobile robot (Figure 2), the displacement of the robot was computed as follows:

$$\begin{aligned} x_{t+1} &= x_t + \frac{V_R + V_L}{2} \cos\theta_t \\ y_{t+1} &= y_t + \frac{V_R + V_L}{2} \sin\theta_t \\ \theta_{t+1} &= \theta_t + \frac{V_R - V_L}{2R}, \end{aligned} \tag{1}$$

where V_R and V_L are the velocities applied to the right and left wheel respectively, R is the radius of a robot and $2R$ is the interval between the wheels. V_R and V_L are set within a continuous range, $[0.0, V_{max}]$, according to the activation of the corresponding output units. The system error was not implemented due to simple analysis.

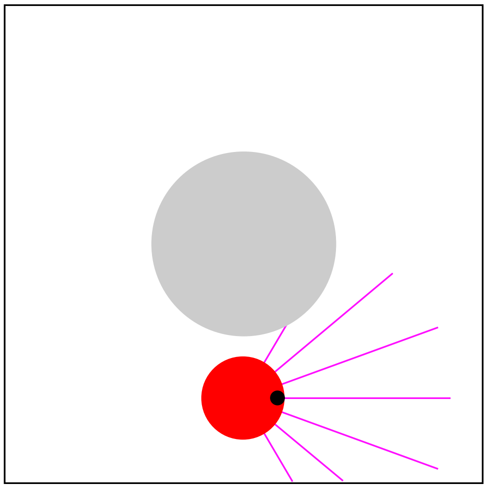


Fig. 1. Experimental setup for a looping maze problem

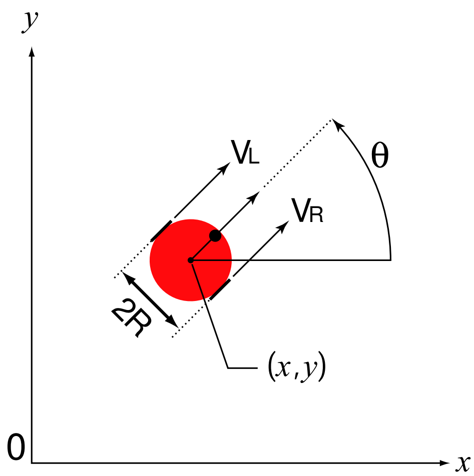


Fig. 2. Simulated model for a mobile robot

III. Fitness Function

The performance measure originally employed by Nolfi & Floreano [3] to be maximized for looping maze was as follows:

$$f = \frac{1}{MaxStep} \sum_{t=1}^{MaxStep} V(1-\sqrt{\Delta V})(1-s_{max}), \tag{2}$$

where $MaxStep$ is maximum step size for a trial, $V = (|V_L| + |V_R|)/2V_{\max}$, $\Delta V = |V_L - V_R|/V_{\max}$, $s_{\max} = \max\{s_j : j = 1, \cdots, N_s\}$ (s_j: the normalized value of the j-th sensor and N_s: the number of sensors). The first component, V, encourages motion, which is called, *MOVE*, the second component, $1-\sqrt{\Delta V}$, encourages the two wheels to rotate in the same direction, *FORWARD* and the third component, $1-s_{\max}$, encourages obstacle avoidance, *AVOID*. In the reference [3], they obtained such looping behaviors as to maximize forward motion while avoiding all obstacles by real-coded GAs. However, it is predictable that such desirable behaviors are not always obtained using Equation 2 due to the differences between experimental setups, e.g. the sizes of environment or arrangements of sensor. In this paper, therefore, those components in Equation 2 are employed as objective functions then the problem was formulated as a multiobjective optimization problem. Considering the physical meaning of each component as well as the number of components, the problem was formulated as two- and three-objective optimization problems as follows.

A. two-objective optimization problem

Instead of the first and the second components in Equation 2, a new objective function, *MOVE FORWARD* was formulated by $V(1-\sqrt{\Delta V})$.

$$\text{Maximize} \quad f_i \qquad (i = 1, 2) \tag{3}$$

$$f_1 = \frac{1}{MaxStep} \sum_{t=1}^{MaxStep} V(1-\sqrt{\Delta V}) \tag{4}$$

$$f_2 = \frac{1}{MaxStep} \sum_{t=1}^{MaxStep} (1-s_{\max}) \tag{5}$$

B. three-objective optimization problem

The three components mentioned above were employed as objective functions, respectively as follows:

$$\text{Maximize} \quad f_i \qquad (i = 1, 2, 3) \tag{6}$$

$$f_1 = \frac{1}{MaxStep} \sum_{t=1}^{MaxStep} V \tag{7}$$

$$f_2 = \frac{1}{MaxStep} \sum_{t=1}^{MaxStep} (1-\sqrt{\Delta V}) \tag{8}$$

$$f_3 = \frac{1}{MaxStep} \sum_{t=1}^{MaxStep} (1-s_{\max}) \tag{9}$$

IV. Neural Controllers

In the reference [7], we proposed to use simply coded evolutionary artificial neural networks (SCEANNs) for robot control in order to shorten the time to evolve robots and illustrated the performance of them using simulated robots. This section describes the details of the SCEANNs.

A. Artificial Neural Networks

Artificial neural networks (ANN) is used with N_s sensory neurons, N_o fully interconnected motor neurons and N_h fully interconnected hidden neurons for a robot's controller.

The output of the i-th neuron at time t is given by:

$$x_i(t) = f(\sum_j \omega_{ij} x_j(t-1)) \tag{10}$$

where ω_{ij} is the connection weight from the neuron j to the neuron i, and $f(x)$ is the output function of neurons, given by the sigmoid function. Namely, their outputs are given by:

$$f(x) = \frac{1}{1+exp(-x/T)} \tag{11}$$

where T is a positive parameter to control the slope of the sigmoid function. The output range is $[0, 1]$.

In general EANN, network's connection weights, firing thresholds for each neuron (the slope when the sigmoid function employed), the architecture of networks and learning rules are evolved [8]. In the SCEANNs, only connection weights are evolved as variables of GAs in order to shorten the length of the genotype. Therefore, the slope of the function, T, is set at 1, which is usually employed in pattern classification.

In [7], we proposed three kinds of architecture. This study employed one of them according to the results obtained in the previous work. The details are described as follows.

B. Simply Coded Evolutionary Artificial Neural Networks (SCEANNs)

The SCEANNs are based on the EANNs originally coded by Floreano[9][10], which was used for evolving real robots. A string is composed of a series of blocks, each block defined for a neuron in hidden and motor neurons (Figure 3). Sensory, hidden and motor neurons are fully connected without coding the presence/absence of a connection from each neuron. Thus, each block is composed of only signs of all the connection weights from sensory, hidden and motor neurons (Figure 4). The synaptic strengths of all existing connections are set at 1. Therefore, the total length of the string is $L = (N_h + N_o)(N_s + N_h + N_o)$ bits, where N_s, N_h and N_o are the number of sensory, hidden and motor neurons, respectively.

V. Computer Simulation

A. Simulation Conditions

The control task used in this study is described in Section II. At the beginning of each trial, a robot was always placed at the same initial position, the bottom left corner, at three kinds of orientation, $\{-45, 0, 45\}^\circ$ (Figure 1). One trial ends either when max steps are performed. The robot behavior is evaluated by using Equation (4)-(5) or (7)-(9). For this experiment, $MaxStep$ is set at 200. A robot's controllers were SCEANNs [7] with 7 sensory neurons, 2 fully interconnected motor neurons and one fully interconnected hidden neurons.

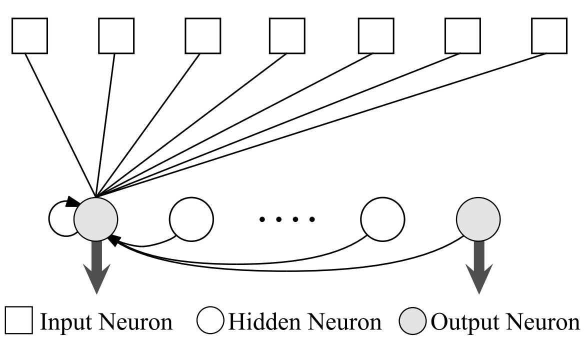


Fig. 3. Architecture of ANNs for 1 block of a string

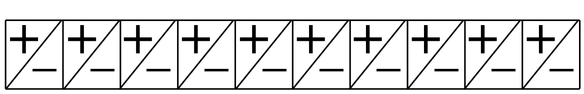


Fig. 4. Genetic representation of one block in the SCEANN

Computer simulations were conducted using populations of size 100. The NSGA-II proposed by Deb *et al.*[11], which is one of the most widely used EMO algorithms, were employed to evolve the string of SCEANNs. The NSGA-II uses standard bit mutation and uniform crossover. The per-bit mutation rate was set at $1/L$ (L: the length of the genotype) and the crossover rate was set at 0.9 following the recommendations given in [11].

A generational model was used. Each run lasted 500 generations. We conducted 100 independent runs.

B. Preference Information and Number of Objectives

As mentioned in Section, this study investigates distribution of preferred solutions in the objective function space obtained from a number of trials in the problem as well as distribution of non-dominated solutions in it. Considering the control task and fitness functions mentioned in Section II, it is intuitively understood that preferred solutions in this problem are the ones which show looping behaviors while avoiding obstacles as fast as possible. In Figure 5, θ_k is defined as a relative angle from the robot's initial position to the current position. At each time step, $\theta_{\max}$ is updated as follows: $\theta_{\max} \leftarrow \theta_k \;\; if \;\; \theta_k > \theta_{k-1}$. Therefore, preferred solutions are defined as the ones which show $\theta_{\max} > 360^\circ$ at the maximum time step.

In this control task, preference information is expressed quantitatively as mentioned above. Perhaps, it may be considered that that information can be employed as one of objective functions. But there is a major conceptual difference between the calculations of Equation (4)-(5), (7)-(9) and θ_k with respect to using internal or external sensors. First of all, Equation (4)-(5) and (7)-(9) are calculated by using only the values of sensors equipped with a mobile robot (e.g., infrared sensors to detect distance and rotary encoders to detect velocities of wheels). On the other hand, the global vision is needed for calculating θ_k, that is, it is assumed that θ_k is calculated by using external sensors which are not equipped with a robot (e.g., an overhead vision camera). Considering autonomy or

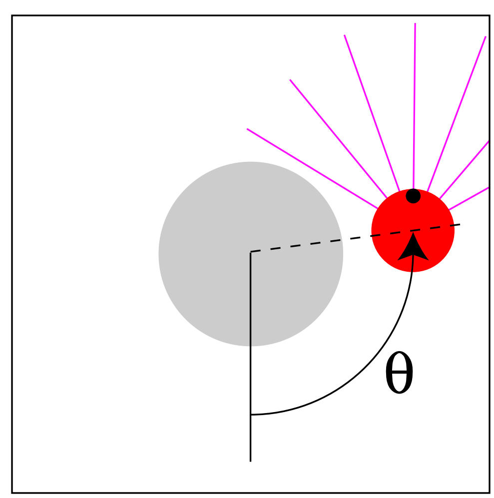


Fig. 5. Relative angle between the initial and current positions of a mobile robot

automation of the whole system including evolutionary computations, it is desirable that objective functions are calculated by using only internal sensors (the similar concept is found in RoboCup Soccer between the settings of small size robot league and middle size robot league [12].). Therefore, θ_k is employed as preference information in this study.

Addition to this, in EMOs, it is reported [13] that the convergence of the obtained non-dominated solutions towards the Pareto-optimal front is declined with a large number of objectives (generally speaking, four or more objectives). In small-objective problem-solving, useful techniques have been proposed so far in the references. Those can be incorporated into our robot systems as well as objective function space can be visualized if the number of objectives is less than four. This is a reason why this problem was formulated as two or three objective optimization problem.

C. Simulation Results

Figure 6 shows the sets of non-dominated solutions (red points) and preferred solutions (green points) of the two-objective optimization problem represented by Equation (4)-(5) obtained at 500th generation in 100 runs, where non-dominated solutions are all the solutions which were ranked best in each run. There is no guarantee that they are Pareto-optimal solutions because the *standard* NSGA-II were employed. But they seem to outline the shape of the Pareto front in the problem. Moreover, we find that the preferred solutions are distributed in a specific region of the non-dominated solutions.

Figure 7 shows the same kind of solutions as Figure 6 of the three-objective optimization problem represented by Equation (7)-(9). It seems that the non-dominated solutions form a Pareto surface in the three-objective function space (Figure 7(d)). The distribution of the non-dominated solutions is difficult to evaluate from the three-dimensional Pareto surface. Therefore, the three projections of the Pareto surface onto the corresponding two-dimensional planes are also provided in Figure 7(a)-7(c). For Figure 7(a) on f_1 and f_3, there are no lower bounds and the non-dominated solutions are distributed widely. For f_2, which shows the competence *FORWARD*, the

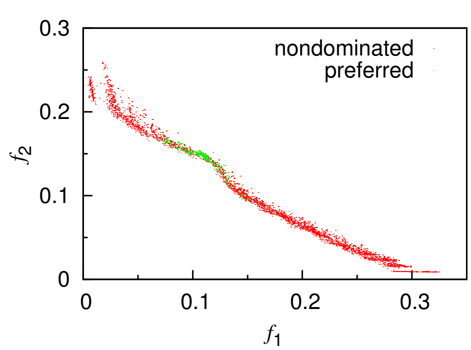


Fig. 6. Distribution of non-dominated solutions and preferred solutions in the two-objective space

non-dominated solutions are not distributed over the range $f_2 < 0.13$ and $0.33 < f_2$ in Figure 7(b)-7(c). This is because if the objective value for f_2, *FORWARD*, is too large, a robot can not avoid any obstacles and if it is too small, a robot mainly rotates around a position or circles even though the objective value for f_1, *MOVE*, is suppressed to a certain degree. As the two-objective optimization problem, we find that the preferred solutions are distributed in a specific region of the non-dominated solutions ($0.13 < f_1 < 0.27$, $0.12 < f_2 < 0.21$ and $0.06 < f_3 < 0.18$) in Figure 7 and 8. It is interesting that the preferred solutions have the small objective values for f_2 in the non-dominated solutions. The preferred solutions show the behaviors where robots avoid any obstacles and loop in the environment. Therefore, robots do not show the desired behaviors in the case that the objective value for f_2 is too large, that is, robots promote *FORWARD* motion as much as as possible.

Figure 9 and 10 show the behavior of a preferred solution of the two-objective and three-objective optimization problem, respectively. We find that the robot completes the loop in the environment while avoiding obstacles. Though it is difficult to distinguish the difference between the behaviors shown in Figure 9 and 10, the behavior of the solution of three-objective optimization problem shows more forward motion than the one of the two-objective optimization problem does. This is because three objective functions were set in Equation (7)-(9) and then Pareto-ranking selection employed in NSGA-II add proper selection pressure to improve values of the objective functions.

VI. Conclusions

This work applied a typical EMO algorithm to the design of ANNs for a controller of a mobile robot in order to investigate distribution of non-dominated and preferred solutions in the objective function space in a problem. The results can be summarized as follows:

- A mobile robot control problem, a looping maze problem, was formulated as a multiobjective optimization problem.
- The shapes of the Pareto front in the problem were almost grasped although the convergence of the obtained non-dominated solutions towards the Pareto-optimal front in real-world problems cannot be evaluated.
- We found in the problem that the preferred solutions are distributed in a specific region of the non-dominated solutions in both the two-objective and three-objective optimization problem.
- The behaviors of preferred solutions show the desired ones in the environment.

These results suggest that EMO algorithms are applicable to the navigation of a mobile robot as well as that an approach to finding a preferred set of Pareto-optimal solution [13] will be useful.

Future work will apply such approaches to multi-robot navigation problems with the same environment in this work.

REFERENCES

[1] K. Deb: *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[2] C. A. C. Coello: An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends, *In Proceedings of the Congress on Evolutionary Computation*, pp. 3–13, 1999.

[3] S. Nolfi and D. Floreano: *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, 2000.

[4] G. Capi: Multiobjective Evolution of Neural Controllers and Task Complexity, *IEEE Transactions on Robotics*, Vol. 23, No.6, pp.1225-1234, 2007.

[5] J. Teo and H. A. Abbass: Automatic Generation of Controllers for Embodied Legged Organisms: a Pareto Evolutionary Multi-objective Approach, *Evolutionary Computation*, Vol. 12, No. 3, pp. 355-394, 2004.

[6] J-H. Kim, Y-H. Kim, S-H. Choi and I-W. Park, Evolutionary multi-objective Optimization in Robot Soccer System for Education, *IEEE Computational Intelligence Magazine*, Vol. 4, No. 1, pp. 31–41, 2009.

[7] Y. Katada and J. Nakazawa, "Investigation of Simply Coded Evolutionary Artificial Neural Networks on Robot Control Problems", In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC2008), pp. 2178–2184, 2008.

[8] X. Yao, Evolving Artificial Neural Networks, In Proceedings of the IEEE, 87(9):1423-1447, 1999.

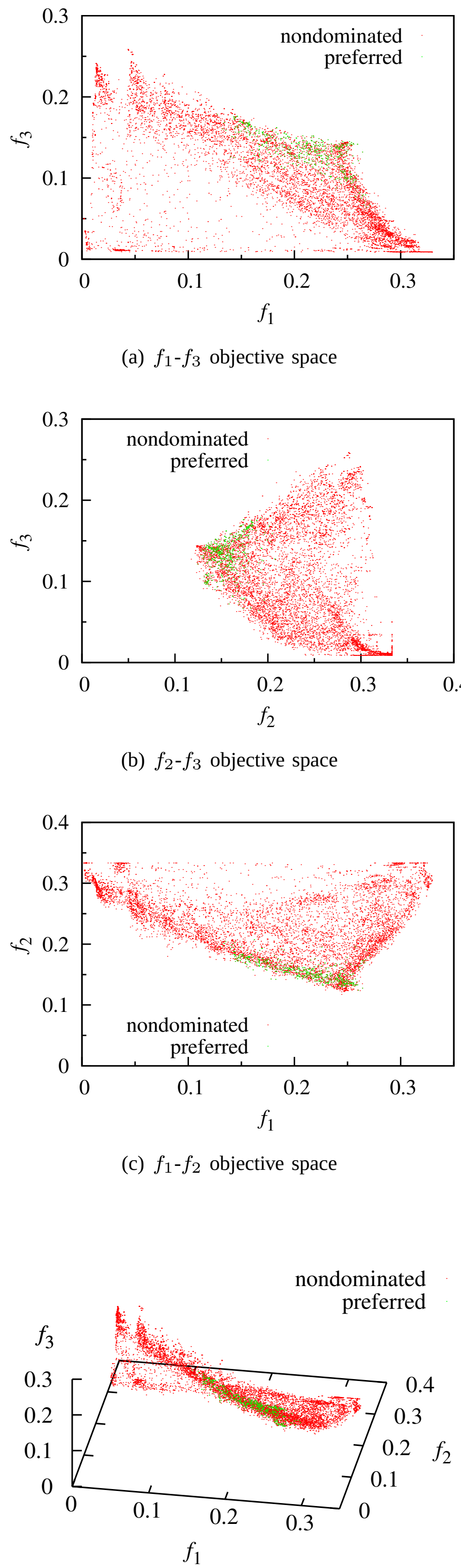
[9] D. Floreano, C. Mattiussi, Evolution of Spiking Neural Controllers, In Gomi, T. (ed.): Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER'01), AAI Books, Springer-Verlag, pp. 38-61, 2001.

[10] J. C. Zufferey, D. Floreano, M. van Leeuwen and T. Merenda, Evolving Vision-based Flying Robots, 2nd International

[11] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan: A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, pp. 182–197, 2000.

[12] URL: http://www.robocup.org/robocup-soccer/

[13] K. Deb and J. Sundar: Reference Point Based Multi-objective Optimization Using Evolutionary Algorithms, *In Proceedings of the 8th annual conference on Genetic and evolutionary computation (GECCO-2006)*, pp. 635–642, 2006.



(a) f_1-f_3 objective space

(b) f_2-f_3 objective space

(c) f_1-f_2 objective space

(d) the three-objective space

Fig. 7. Distribution of non-dominated solutions and preferred solutions in the three-objective space

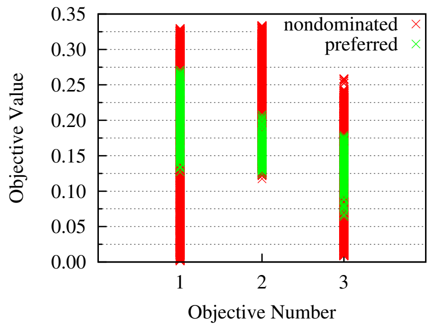


Fig. 8. Objective value of solutions

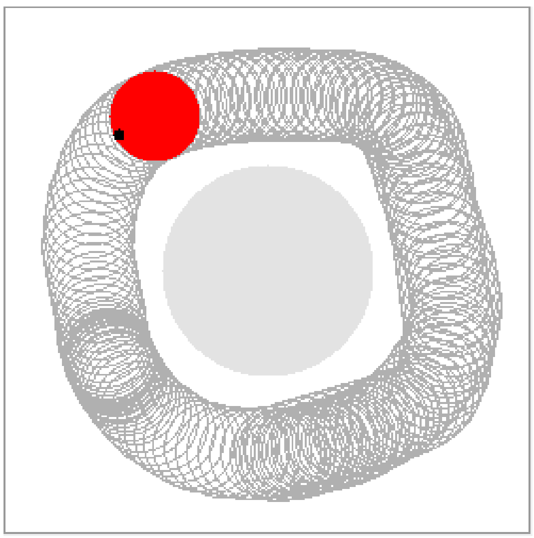


Fig. 9. Behavior of a preferred solution of the two-objective optimization problem

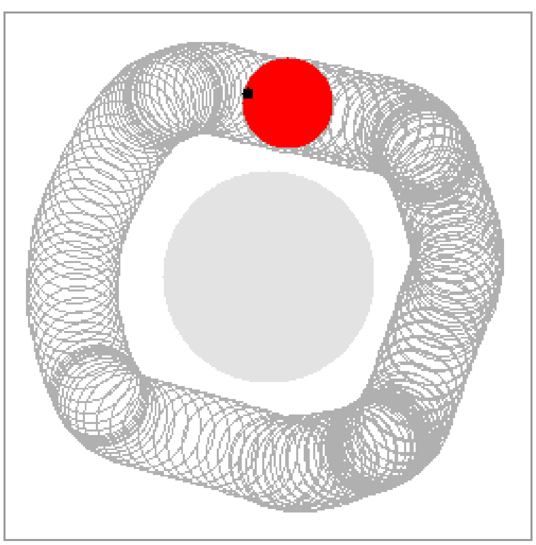


Fig. 10. Behavior of a preferred solution of the three-objective optimization problem